

빠른 리젝션과 고효율 특징선택을 이용한 빠른 Viola-Jones 물체 검출기

박병주*, 이재홍*, 이광호**

*한밭대학교 컴퓨터공학과

** (주)에어포인트

e-mail: dinobei89@gmail.com

Fast Viola-Jones Object Detector using Fast Rejection and High Efficient Feature Selection

Byeong-Ju Park*, Jae-Heung Lee*, Gwang-Ho Lee**

*Dept of Computer Engineering, Hanbat National University

**Airpoint Co.,Ltd.

요 약

본 연구에서는 기존의 Viola-Jones 물체 검출 프레임워크를 개선하여 하나의 특징 당 더 높은 효율을 가지며 검출대상이 아닌 서브 윈도우들을 더 빠르게 제거하는 학습 알고리즘을 제안한다. 학습의 결과로 생성된 물체 검출기는 서브윈도우를 특정 임계값까지 빠르게 제거하기 때문에 서브윈도우당 계산수가 줄어든다. 기존의 Viola-Jones 물체 검출기와 동일한 프레임워크이므로 인식성능에는 영향을 주지 않는다. MIT-CMU 테스트 집합에 대해서 서브윈도우당 특징 계산 횟수를 측정하였으며 기존 계산 횟수의 57%로 줄어들어 검출 속도가 약 71% 향상됨을 확인하였다.

1. 서론

Viola-Jones 물체 검출 프레임워크는 실시간으로 높은 물체 검출률을 달성하기 위해 제안된 최초의 물체 검출 프레임워크이다. 보행자나 얼굴, 차량, 번호판, 손 등을 검출하기 위한 응용사례가 있으며 적용 결과 Viola-Jones 알고리즘은 다른 검출 알고리즘에 비해 검출 속도도 빠르면서 검출률도 비슷하거나 우수한 것으로 알려져 있다. Viola-Jones 알고리즘의 핵심은 초기에 정한 포지티브 샘플들의 패턴과 그 외의 모든 패턴을 구분하는 것으로 볼 수 있다. 특정 서브윈도우에서 나올 수 있는 패턴은 검출하려고 하는 물체의 모든 패턴에 비해 굉장히 많기 때문에 FPR (False Positive Rate)을 최대한 낮게 학습하는 것이 관건이다. 다만, 다양한 패턴의 영상에서는 오인식할 확률이 높아지는 단점이 있다.

기존의 Viola-Jones 물체 검출 프레임워크는 적분 이미지를 사용하여 Haar-like feature를 빠르게 계산할 수 있고 빠른 리젝션을 위해 cascade 형태로 학습한다. 하지만 인식성능을 더욱 높이려는 연구를

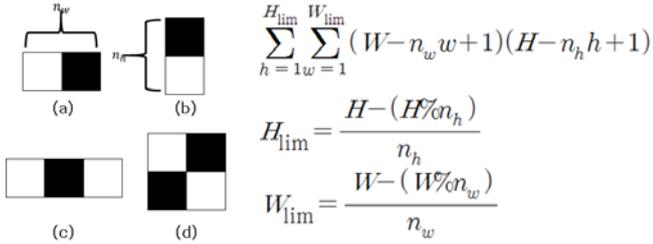
하면서 불가피하게 연산량이 늘어나게 되었고 이를 적용하면서도 속도를 향상시킬 수 있는 연구가 필요하게 되었다.

본 연구에서는 초기에 보다 빠른 리젝션을 통해 서브윈도우당 평균 특징 계산 횟수를 줄였고, 가장 좋은 효율의 Haar-like feature를 선택하여 각각의 약분류기당 가장 많은 서브윈도우들을 제거하도록 하였다. 이러한 두 가지 개념을 추가하여 동일한 학습 데이터로 보다 빠른 검출기를 학습하는 개선된 Viola-Jones 물체 검출 알고리즘을 제안한다.

2. Viola-Jones 물체 검출기 관련 연구

Viola-Jones 알고리즘에서 사용하는 Haar-like feature의 overcomplete set은 임의의 비율로 정교하게 샘플링된 위치를 가진 특징들을 제공[1]하기 때문에 풍부한 특징 표현법이며, 특징 계산 속도가 굉장히 빠른 특징을 갖고 있다. 서브윈도우 크기에 따른 Haar-like feature의 overcomplete set의 수는 그림 1의 수식으로 계산할 수 있다. 그러나 Haar-like feature는 경계선 정보만을 이용하기 때문에 성능 향상에 한계가 있다.

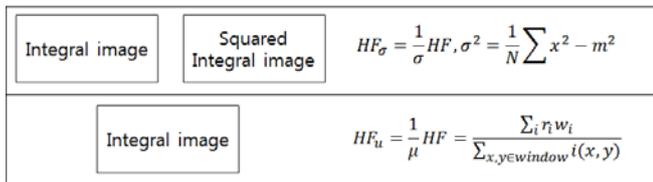
※ 본 연구는 교육과학기술부와 한국연구재단의 지역 혁신인력양성사업으로 수행된 연구결과임



(그림 1) Haar-like feature와 overcomplete set의 수

또 다른 한계점은 정규화이다. Haar-like feature는 정규화가 필수적인데, 기존의 분산정규화를 이용하면 영상의 밝기 변화가 작은 영역에서 오검출이 증가[2]하게 되며, 정규화를 위해 별도의 적분영상을 더 생성해야하기 때문에 처리 속도에 적지 않은 영향을 미치게 된다.

이를 개선하기 위해 현재까지 연구된 Viola-Jones 물체 검출기 관련 연구 주제는 크게 속도 또는 검출 성능 향상의 두가지로 볼 수 있다. 인식 성능 향상을 위해서 Extended Haar-like feature Set[3]이나 Joint Haar-like feature[4], Asymmetric Haar-like feature[5]와 같이 overcomplete set을 추가하거나 변경하는 연구가 있었고, 이들 연구의 공통된 목적은 더 풍부한 표현력을 가진 특징을 이용해서 더 좋은 검출기를 만드는 것이다. 또, 기존의 정규화 방법 대신 연산량을 줄인 Haar-like feature 정규화에 대한 연구[2]도 있다. 이 연구에서 제안하는 정규화는 그림 2와 같다. [2]의 결과로 영상의 밝기 변화에 대해서 강건한 성능을 보이며 별도의 적분 이미지 생성을 필요로 하지 않아 속도가 개선되었다.



(그림 2) 기존의 분산 정규화 방법과 개선된 Haar-like feature 정규화 방법[2]

다음으로 속도 향상을 위해서 기존의 스캔 방식인 슬라이딩 윈도우 대신에 PSO를 적용하여 계산할 서브윈도우 수를 줄인 연구[6]와 얼굴검출기의 성능을 높이기 위해 멀티코어 환경에서 피부색으로 스캔 영역을 한정하고 알고리즘 자체를 병렬화한 연구[7]가 있다.

3. 본 논문의 개선된 Viola-Jones 물체 검출기

본 논문에서 개선한 내용은 크게 세 가지로 나눌 수 있다. 첫 번째는 네거티브 샘플의 평가기준 변경이다. 기존의 Viola-Jones 물체 검출 알고리즘에서는 네거티브 샘플을 random하게 선택하여 이에 종속된 Adaboost 학습을 한다. random하게 선택된 네거티브 샘플들이기 때문에 만약 현재 남아있는 서브윈도우 중에서 분류에 특징이 많이 필요한 샘플이 다수 선택되면 최종적으로 만들어진 검출기의 연산량이 많아져 비효율적일 수 있다. 또한 각 스테이지의 FPR을 계산하는데 있어서 random하게 선택된 네거티브 샘플들로 평가하게 되면, 선택된 샘플에 종속적으로 FPR이 결정되기 때문에 전체를 대상으로 평가했을 때와 비교해서 편이하게 다른 결과를 얻을 수도 있다. 실제로 본 연구에서는 기존 학습 결과를 테스트해보았고, 표 1과 같이 초기 스테이지의 성능에 따라 평균 계산횟수가 차이나서 검출기의 속도가 최대 2.09배 차이나는 것을 확인하였다.

<표 1> MIT-CMU 테스트 집합에 대한 기존의 Viola-Jones 물체 검출기 테스트 결과

학습 횟수	1	2	3
FPR	5.6E-07	1.85E-06	2.84E-06
평균 계산 횟수	3.85	6.29	8.48
평균 계산 속도	83.54ms	121.47ms	174.98ms

따라서 random하게 네거티브 샘플을 선택하되, 검출할 물체가 포함되지 않은 영상 집합에 대해 각 스테이지의 FPR과 효율을 측정하였다.

두 번째는 빠른 리젝션을 위한 Adaboost의 종료 조건 변경이다. 기존의 학습 알고리즘에서는 스테이지당 목표 FPR인 f 가 입력 파라미터로 들어가서 (1)과 같이 이전 스테이지보다 f 만큼 더 filter out 능력이 좋아져야만 하나의 스테이지 학습이 종료되었다. 이렇게 되면 빠른 리젝션을 하지 못해 비효율적인 분류기가 생성 될 여지가 있다.

$$F[n-1]*f > F[n] \tag{1}$$

$$F[n-1] > F[n] \tag{2}$$

따라서 선택된 약분류기의 수와 상관없이 효율이 가장 좋을 때 빠르게 리젝션 할 수 있도록 (2)의 조건을 사용했다. (2)의 조건으로 변경하여 학습하면 스테이지가 매우 많아지게 되는데, cascade 구조에서는 스테이지가 많아질수록 최종 검출률이 떨어지

게 된다. 따라서 filter out 임계값($threshold_{filterout}$)을 정의해서 그 임계값 전까지만 (2)의 조건을 사용하도록 설계하였다. 최종적으로 변경된 Adaboost 종료조건은 그림 3과 같다.

```

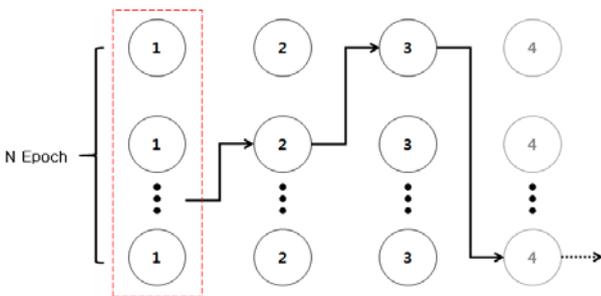
F[n] = (Evaluate FPR for NFIS)
if(1.0 - F[n - 1] ≥ thresholdfilterout && F[n] < f * F[n - 1])
    break;
else if(F[n] < F[n - 1]) break;
    
```

(그림 3) 빠른 리젝션을 위한 Adaboost 종료조건

세 번째는 가장 높은 효율의 스테이지를 선택하기 위한 변경된 학습 구조이다. 각 스테이지 학습 중 random하게 선택된 네거티브 샘플에 따라 가장 많은 FPR의 감소를 보이는 초기 스테이지의 학습 결과가 달라질 수 있다.

$$\frac{1 - FPR}{\text{Number of Weak classifier in Stage}} \quad (3)$$

따라서 학습 중 발견한 가장 좋은 효율의 스테이지를 선택하기 위해 퍼셉트론의 포켓 알고리즘과 같이 세대(Epoch)를 두었다. 즉, 각 세대 중 최고의 효율을 발휘하는 스테이지를 선택하고 몇 번의 세대가 진행되는 동안 변화가 없으면 현재 스테이지의 학습을 마치도록 변경되었다. 효율 계산을 위해 (3)을 사용하였으며 변경된 학습 구조는 그림 4와 같다.

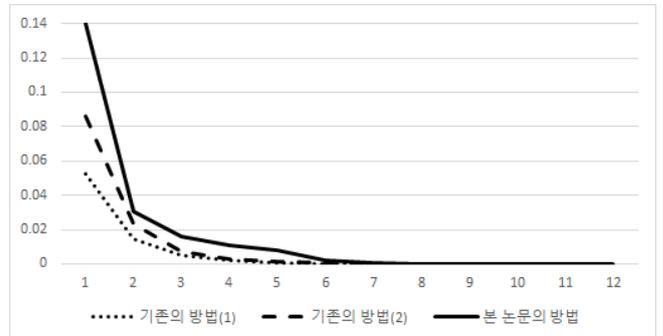


(그림 4) 가장 높은 효율의 스테이지를 선택하기위해 변경된 학습 구조

4. 실험 및 고찰

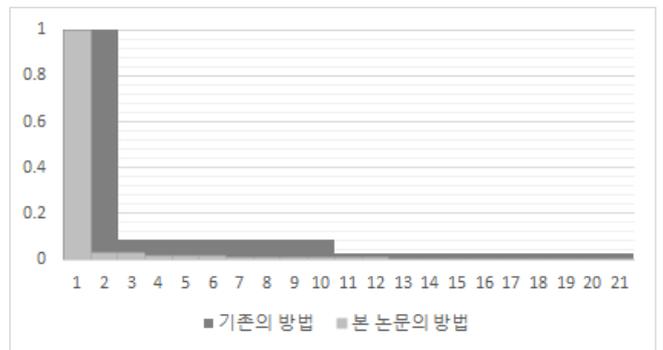
기존의 다양한 응용 중 얼굴 검출을 바탕으로 실험하였다. LFW (Labeled Faces in the Wild) 얼굴 데이터베이스에서 정렬된 데이터를 이용해 직접 제작한 7406개의 얼굴 샘플과 random하게 촬영된 1,307장의 NFIS (Non-face image set)를 학습 데이터로 사용하였다. 기존의 Viola-Jones 얼굴 검출기

[1]의 연구 결과를 토대로 scale factor와 d, f는 각각 1.25, 0.99, 0.3으로 두고 학습하였고, 특징의 정규화는 [2]의 방법을 사용하였다. 각 스테이지별 FPR은 그림 5와 같다.



(그림 5) 스테이지별 FPR

그림 5의 그래프를 보면 스테이지별 수렴 속도가 기존보다 느리지만 표 2를 보면 본 연구의 방법으로 학습한 검출기의 스테이지별 약분류기 수가 더 적다는 것을 알 수 있다. 따라서 약분류기별로 FPR을 살펴보면 평균적으로 서브윈도우당 특징 계산 횟수가 줄어든 것을 알 수 있다. 그림 6은 약분류기별 오검출률을 보여주는 그래프인데 줄어든 계산 횟수를 면적으로 확인할 수 있다.



(그림 6) 약분류기별 오검출률

NFIS와 포지티브 샘플로 학습하여 얻은 검출기는 NFIS에 대한 약분류기의 평균 계산횟수(nc_m)를 수식으로 계산할 수 있다. 즉, n_x 가 스테이지 x 의 약분류기 수이고 f_{x-1} 이 스테이지 $x-1$ 에서의 FPR이라면 (4)와 같이 표현된다.

$$nc_m = \sum_{x=1}^N n_x f_{x-1}, f_0 = 1 \quad (4)$$

(4)의 수식으로 평균 계산 횟수를 비교했을 때 기존 방법의 평균 계산횟수는 2.96이었으며 본 논문의 방법은 1.71로 기존에 비해 약 57.67%로 줄어들었다. 동일하게 MIT-CMU 테스트 집합으로 비교해봤을 때는 56.62%로 측정되었다. 다양한 패턴의 NFIS로 학습하였기 때문에 임의의 테스트 집합으로 비교해봐도 유사한 결과를 얻을 수 있었다.

<표 2> 기존의 방법과 본 논문의 학습 결과 비교

	기존의 방법		본 논문의 방법	
	FPR	N _{wc}	FPR	N _{wc}
stage1	0.086021	2	0.139901	1
stage2	0.023486	8	0.030257	2
stage3	0.007253	14	0.015886	3
stage4	0.002726	21	0.010729	3
stage5	0.001114	29	0.008003	3
stage6	0.000403	36	0.002392	22
stage7	0.000150	37	0.000635	34
...

MIT-CMU 테스트 집합을 기준으로 이전의 검출 성능과 이후의 검출 성능을 85% 전후로 거의 동일하게 맞춘 뒤 결과를 측정하였다. 기존 알고리즘의 학습 결과를 테스트한 것은 표 1이며 본 논문에서 제안하는 알고리즘의 학습 결과를 테스트한 것은 표 3이다. 각각 가장 좋은 결과일 때를 비교해보면 약 71% 검출 속도가 향상된 것을 알 수 있다. 검출 속도의 개선 정도는 사용하는 특징의 계산량에 따라 다르게 측정되겠지만, 기본적으로 계산량이 증가하면 실제 검출 속도 향상 폭은 더욱 증가할 것이다.

표 3 MIT-CMU 테스트 집합에 대한 본 논문의 방법을 적용한 검출기 테스트 결과

학습 횟수	1	2	3
FPR	9.91E-08	3.30E-08	6.70E-08
평균 계산 횟수	2.20	2.20	2.18
평균 계산 속도	49.85ms	49.43ms	48.83ms

본 연구의 결과는 기존의 연구와 독립적이기 때문에 다른 연구결과를 접목하면 보다 큰 성능향상이 가능할 것으로 보인다. 예를 들어 다른 특징을 사용하여 인식 성능을 높이거나 스캔 방식을 변경해서 속도를 더욱 향상시킬 수 있다.

5. 결론

본 논문에서는 빠른 리젝션과 고효율 특징 선택을 이용한 빠른 Viola-Jones 물체 검출기를 제안하

였다. 네거티브 샘플을 random하게 선택하여 각 스테이지를 학습할 때 생기는 문제점을 해결하기 위해 FPR을 평가하는 기준을 변경하였고 세대수 개념을 도입한 학습 구조를 제시하였다. 그리고 가장 좋은 효율의 스테이지를 선택하도록 Adaboost 종료 조건을 변경하였다.

성능 비교를 위해 기존의 방법과 본 논문에서 제안하는 방법에 대해 각각 얼굴 데이터로 학습하였고, MIT-CMU 테스트 집합을 기준으로 학습 결과를 얻었다. 본 논문에서 제안하는 방법을 적용한 결과 검출률에는 영향을 주지 않았으며, 가장 좋은 결과를 기준으로 비교했을 때 서버윈도우당 평균 계산 횟수가 57%로 감소하였고 검출 속도는 평균 71% 향상되었다.

참고문헌

[1] Paul Viola, Michael J. Jones "Robust Real-Time Face Detection" International Journal of Computer Vision, vol.57, no.2, pp.137-154, May. 2004

[2] Ki-Yeong Park, Sun-Young Hwang "An Improved Normalization Method for Haar-like Features for Real-time Object Detection" The Korean Institute of Communications and Information Sciences, vol.36, no.8, pp.505-515, Aug. 2011

[3] Chang Chai, Yulin Wang "Face Detection Based on Extended Haar-like Features" 2nd International Conference on Mechanical and Electronics Engineering, vol.1, no., pp.442-445, Aug. 2010

[4] Takeshi Mita, Toshimitsu Kaneko, Osamu Hori "Joint Haar-like Features for Face Detection" Tenth IEEE International Conference on, vol.2, no., pp.1619-1626, Oct. 2005

[5] Geovany A. Ramirez, Olac Fuentes "Multi-Pose Face Detection with Asymmetric Haar Features" Applications of Computer Vision, 2008. WACV 2008. IEEE Workshop on, vol., no., pp.1-6, Jan. 2008

[6] Ammar W Mohemmed, Mengjie Zhang, Mark Johnston "Particle Swarm Optimization Based Adaboost for Face Detection" Evolutionary Computation, 2009. CEC '09. IEEE Congress on, vol., no., pp.2494-2501, May. 2009

[7] Hong-Hee Kim, Jae-Heung Lee "An Efficient Face Detection Method Using Skin Color Information and Parallel Processing in Multi-Core SoC" Journal of IKEEE, vol.16, no.4, pp.375-381, Dec. 2012